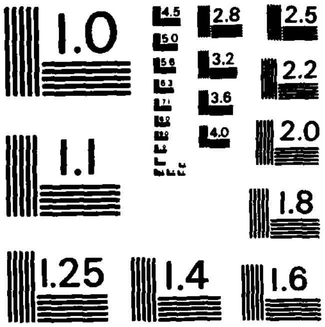
AD-A160 042   INTEGRATING SYNTAX SEMANTICS AND DISCOURSE DARPA   1/1
NAUTRAL LANGUAGE UNDERSTANDING PROORAM(U) SYSTTM
DEVELOPMENT CORP PAOLI PA L HIRSCHMAN 13 SEP 85
UNCLASSIFIED N00014-85-C-0012   F/G 5/7   NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

2

INTEGRATING SYNTAX, SEMANTICS, AND DISCOURSE DARPA NATURAL LANGUAGE UNDERSTANDING PROGRAM

R&D STATUS REPORT
SDC -- A BURROUGHS COMPANY

SDC
P.O.Box 517
Paoli, Pa. 19301

ARPA ORDER NUMBER: 5262
PROGRAM CODE NO. NR 049-602 dated 10 August 1984 (433)
CONTRACTOR: System Development Corporation   CONTRACT AMOUNT: $683,105
CONTRACT NO: N00014-85-C-0012
EFFECTIVE DATE OF CONTRACT: 4/29/85   EXPIRATION DATE OF CONTRACT: 4/29/87
PRINCIPAL INVESTIGATOR: Dr. Lynette Hirschman   PHONE NO. (215) 648-7554

SHORT TITLE OF WORK: DARPA Natural Language Understanding Program

REPORTING PERIOD: 4/29/85 - 8/31/85

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

This document presents

1. Description of Progress

Figure 1 shows an overview of the SDC/NYU natural language system. The following paragraphs describe the progress during the first quarter associated with the various system components. A major

Contents: grammar, noun phrase resolution, clause semantics, facilities.

1.1. Grammar

The grammar is being extended to coverage the wider range of constructions found in the CASREPs material. Current efforts are focussed upon both the BNF and restriction components of the grammar. These changes include:
(1) simplification of the verbal auxiliary system, with a corresponding simplification of the yes-no question rule; (2) a more fine-grained treatment of infinitival complements, which should permit selection restrictions to apply more accurately; and (3) a fuller treatment of the internal structure of noun and adjective phrases. The restriction component of the grammar is currently being developed to accommodate these changes, since they would would otherwise greatly increase the number of parses generated by the grammar.

In addition, a conjunction mechanism has been added to handle conjoining both within noun phrases and within assertions. This component has been tested, but not yet integrated into the semantic component.

1.2. Noun Phrase Resolution

The noun phrase resolution module consists of two components, noun phrase semantics and reference resolution.

Noun phrase semantics is called by the parser during the parse of a sentence, after each noun phrase has been parsed. It performs two functions. The first function is to assist the parser by informing it whether or not the parsed noun phrase is semantically acceptable in the current domain. For example, in the sentence, *field engineer replaced disk drive at 11/2/0800*, *disk drive at 11/2/0800* is a syntactically acceptable noun phrase, (as in *participants at the meeting*). However, it is not semantically acceptable in that *at 11/2/0800* is intended to designate the time of the replacement, not a property of the disk drive. Noun phrase semantics will inform the parser that the noun phrase is not semantically acceptable, and the parser can then look for another parse. In order for this capability to be fully utilised, however, an extensive set of domain-specific rules about semantic acceptability is required. At present we have only the minimal set used for the development of the basic mechanism. [For example, in the case described here, *at 11/2/0800* is excluded as a modifier for *disk drive* by a rule that permits only the name of a location as the object of *at* in a prepositional phrase modifying a noun phrase. In the form of a Prolog clause this rule can be stated as follows.

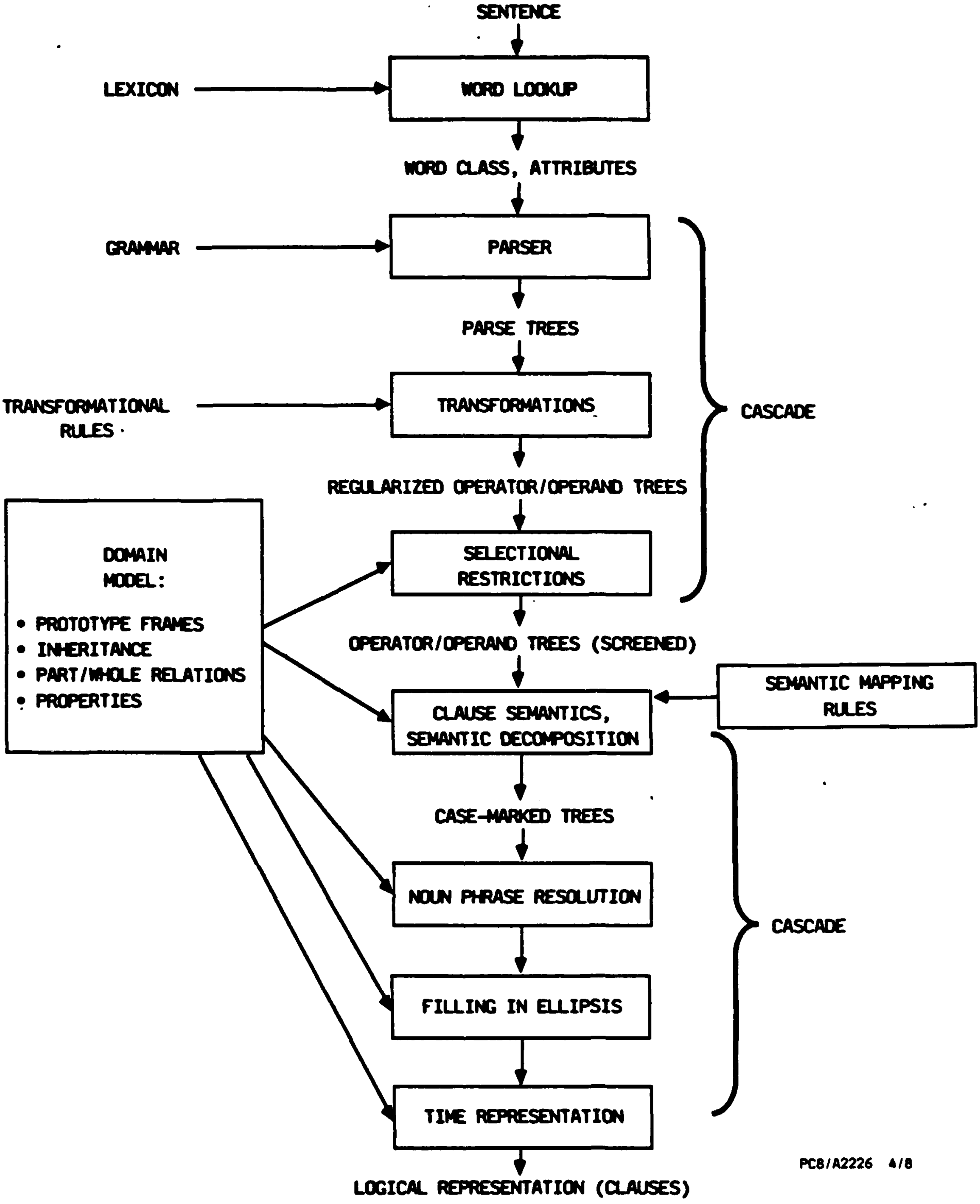
```
at(X,_):-
   isa2(X,location).
```

The second function of noun phrase semantics is to create a semantic representation of the noun phrase, which will later be operated on by reference resolution. For example, the semantics for *the bad disk drive* would be represented by the following Prolog clauses.

```
hasname(disk^drive,X).
bad(X).
def(X).     % that is, X was referred to with a full, definite noun phrase,
full_np(X). % rather than a pronoun or indefinite noun phrase.
```

The second component, reference resolution, is currently integrated with clause semantics. The functions of clause semantics are to create a semantic decomposition for the verbs in a text and to instantiate the semantic roles of the predicates in the decomposition. When clause semantics is ready to instantiate a semantic role, it calls reference resolution, which provides a referent, e.g., [drive1], as the filler for the role. Control then returns to clause semantics, which checks the selectional restrictions on the verb to determine whether or not the proposed referent is semantically appropriate. If it is not, reference resolution is called again; otherwise clause semantics continues instantiating semantic roles. Reference resolution not only provides referents, but also performs other pragmatic functions, such as associating parts with what they are part of. For example, something referred to as *the motor* can be associated with a previously mentioned disk drive that it is part of. In order to perform this association, reference resolution makes heavy use of the knowledge represented in the domain model. A version of Sidner's (1979) focusing algorithm is used for this function, as well as for pronoun resolution. Recently

SDC SYSTEM STRUCTURE



SENTENCE

LEXICON

WORD LOOKUP

WORD CLASS, ATTRIBUTES

GRAMMAR

PARSER

PARSE TREES

TRANSFORMATIONAL RULES

TRANSFORMATIONS

REGULARIZED OPERATOR/OPERAND TREES

DOMAIN MODEL:

- PROTOTYPE FRAMES
- INHERITANCE
- PART/WHOLE RELATIONS
- PROPERTIES

SELECTIONAL RESTRICTIONS

CASCADE

OPERATOR/OPERAND TREES (SCREENED)

CLAUSE SEMANTICS, SEMANTIC DECOMPOSITION

SEMANTIC MAPPING RULES

CASE-MARKED TREES

NOUN PHRASE RESOLUTION

FILLING IN ELLIPSIS

TIME REPRESENTATION

CASCADE

LOGICAL REPRESENTATION (CLAUSES)

PC8/A2226 4/8

the reference resolution module has been augmented with the capability of resolving pronouns that refer to events, as in *Field engineer replaced disk drive. It fixed the system.* In this example, *it* would be resolved as meaning *the fe's replacement of the disk drive.*

1.3. Clause Semantics

Our initial work was done on a Maintenance Report domain consisting of SDC maintenance reports. This corpus allowed us to begin developing the Clause Semantic component for an actual maintenance domain, while negotiations were being carried out to determine the actual domain for the deliverable system.

The texts analysed are actual maintenance reports as they are called into the Burroughs Telephone Tracking System by field engineers and typed in by telephone operators. These reports give information about a customer who has a problem, specific symptoms of the problem, any actions taken by the field engineer to try to correct the problem, and the success or failure of such actions. The goal of the text analysis is to automatically generate a data base of maintenance information that can be used to correlate customers to problems, problem types to machines, and so on.

The first step in building a domain model for maintenance reports was to build a semantic net representation of the machine involved. The machine in the example text given below is a B4700. The possible parts of a B4700 and the associated properties of these parts can be represented using 4 basic predicates: **system, haspart, hasprop, hasvalue.** For example the system itself is indicated by **system(b4700).** The main components of the system - cpu, power_supply, disk, printer, peripherals, etc. - are indicated by **haspart** relations, such as **haspart(b4700,cpu), haspart(b4700,power_supply), haspart(b4700,disk),**,etc. These parts are themselves divided into subparts which are also indicated with haspart relations, such as **haspart(power_supply, converter).** Parts can have properties associated with them, i.e., **hasprop(fuse,amp),** which can also have values, i.e., **hasvalue(amp,70).** This method of representation results in a general description of a type of computer system. Specific machines represent *instances* of this general representation. When a particular report is being processed, **hasname** relations are created to associate the specific computer parts being mentioned with the part descriptions from the general machine representation. So a particular B4700 would be indicated by predicates such as these: **hasname(b4700,system1), hasname(cpu,cpu1), hasname(power_supply, power_supply1),** etc.

To represent the information conveyed in the report, two other types of representation are needed as well, **causal events** and **script events.** A given event can be both a causal event and a script event. The representation includes a name, i.e., event1, a predicate decomposition of the event, and a time argument. The time can be explicit, such as 03-14/1700, or relative, as in **before(event2).** The **causal events** are used to represent the underlying problems in the computer system. For instance, a failure of the power_supply can cause a failure of the entire system. These causal chains are referred to explicitly in the text, as in *System down with solid power failure.* The causal events are produced by doing a semantic analysis of the sentence fragments of the report, with each syntactic fragment generally corresponding to a causal event. The semantic processing technique used is Inference-Driven Semantic Analysis. During the first quarter, we analysed the maintenance texts and developed semantic rules for the basic maintenance verbs; we have also begun to investigate the use of Inference-Driven Semantic Analysis techniques for the analysis of nominalised verbs, as in *The field engineer performed the alignment of the disk* vs. *The field engineer aligned the disk.*

Script events represent the sequence of actions for a maintenance call. The report begins with the customer call, so this is the first script event. The description of the problem is usually the second event and so on. The assignment of events to the script events will be produced after the text has been analysed and the causal event representation has been established. Script events have not been implemented yet.

1.4. Facilities

Most of the work that we have done on the Symbolics machines has been preparatory for software development. Several people have been studying the machine and learning how to incorporate it into our working environment. In addition, the porting of already-developed software from our Vax has already begun.

We have networked together the Symbolics machines with our Vaxen. Both machines can copy files to each other, and the Symbolics machines can use the Vaxen as a file server, as well as remotely executing commands on the Vax. We have also implemented a scheme to allow the Symbolics machines to use the printers connected to the Vax for hardcopy and screen dumps. A lot of small customisations and modifications have been made to tailor the machines to fit our own tastes for a friendly development environment.

We have also been working out mechanisms to allow several people to work on the same core programs and maintain individual modifications, as well as incorporating techniques for controlling the complexity of large programs. A reasonable backup system has also been investigated.

The people who have been accumulating experience working on the Symbolics machines have started to transfer their knowledge to other users. Some of the Natural Language System has been successfully ported over to the Symbolics machine; in the process, we have uncovered numerous bugs in the Symbolics Prolog implementation. Current progress is hampered by lack of support by Symbolics for its Prolog implementation.

2. Change in Key Personnel

The project is currently fully staffed by the following people (approximate percent of time indicated as well).

Lynette Hirschman	Sr. Staff Research Scientist	33%
Martha Palmer	Staff Research Scientist	17%
Deborah Dahl	Research Scientist	50%
Marcia Linebarger	Research Scientist	33%
Karen Wieckert	Research Scientist Assoc.	100%
John Dowding	Research Scientist Assoc.	100%

3. Summary of Substantive Information from Meetings and Conferences

3.1. Professional Computational Linguistics Meetings

ACL '85 (July 8-12, University of Chicago)

The 23rd meeting of the Association of Computational Linguistics was attended by John Dowding and Karen Wieckert. Dowding and Wieckert attended tutorials on ??, as well as attending sessions on ...

IEEE Logic Programming Symposium (July 15-18, Boston, Mass.)

This Symposium was sponsered by the IEEE Technical Committee on Computer Languages, and was attended by John Dowding and Lew Norton. Dowding focused on sessions relating to parsing issues.

3.2. SDC/NYU Meetings

SDC/NYU Meeting #1 (February 21, New York University)

Lynette Hirschman, Martha Palmer, Deborah Dahl, John Dowding and Karen Wieckert attended a meeting with Grishman, Ksiesuk and Nhan at NYU. Topics of discussion included noun phrase semantics (Dahl), chart parsing (Gawron), conjunction parsing (Hirschman), frame representation (Ksiesyk), and clause semantics (Palmer).

SDC/NYU Meeting #2 (March 22, Paoli, PA).

Ralph Grishman, Mark Gawron, Tomass Ksiesyk, and Ngo Thanh Nhan came to Paoil for a meeting. Martha Palmer gave a talk on clause semantics, John Dowding and Karen Wieckert gave a talk on lexical lookup and entry procedures, Tomas Ksiesyk gave a talk on a frame representation language, and Mark Gawron gave a talk on using lamdba conversion to build an intermediate semantic representation.

SDC/NYU Meeting #3 (April 15, Paoli, PA)

Staff from SDC and NYU met again to work out a presentation at the DARPA Technology Review Panel meeting in Hawaii. We also pursued discussions on lamdba conversion, the relationship between frames and clause semantics, and the overall organisation of the system.

SDC/NYU Meeting #4 (June 27 - New York University)

On this trip John Dowding and Marcia Linebarger from SDC went to NYU specifically to see Mark Gawron at NYU to discuss parsing issues.

3.3. DARPA Meetings

DARPA Natural Language Technology Review Panel Meeting (May 1-3, Hawaii)

The first Natural Language Technology Review Panel (NLTRP) meeting was held at CINCPACFLT, Pearl Harbor, Hawaii. This meeting included the major natural language contracts (BBN, ISI, SDC, NYU), as well as Texas Instruments, the Expert Systems contract. The overall structure of the Battle Management Program was reviewed, and we then focused on the natural language components and their integration. At this meeting, we held side meetings with CINCPACFLT personnel and various DARPA personnel and consultants to choose a domain for text processing. The outcome of these meetings was a proposal to use an existing collection of CASREP texts describing failures of the starting air compressor. The drawback to this material was its very narrow focus, which made its relation to the broader Force Readiness Expert System (FRESH) less clear; however, the conclusion was that it would be too burdensome to model any broader domain, for the initial system.

DARPA Meeting of Natural Language Contractors, (May 22-23, Roslyn, VA)

This meeting served to introduce and organise the seven NL contractors into two groups: SDC, NYU and SRI for text processing; BBN, ISI, U. Mass and Penn for user interfaces. The meeting also raised questions around integration of components into a final system and criteria for measuring success of the overal system and its components.

3.4. Symbolics Lisp User's Group (SLUG) Meetings

SLUG Meeting (June 3 & 4,- San Francisco, CA)

SLUG '85 was the organisational meeting for the Symbolics LISP Users Group, and was attended by John Dowding and Karen Wieckert. Useful information, including details of the design of Symbolics Prolog, was obtained both from Symbolics officials, and other experienced Symbolics users.

MAD-SLUG Meeting (June 12, Princeton University)

This meeting of the Mid-Atlantic Division of SLUG was attended by John Dowding, Charles Kosloff, and Chris Andrews. These quarterly meetings will be hosted by different Universities and Corporations. At this meeting, there were several guest speakers from Symbolics, and the details of organising a library of free software were worked out.

4. Problems Encountered and/or Anticipated

The major problem to date concerns Prolog for the Symbolics Machine. SDC has two machines installed: one purchased by SDC and one GFE. At the moment, only the SDC machine has Prolog. The Government-furnished machine has no Prolog, due to licensing problems between DARPA, ISI and Symbolics. In addition, the Prolog furnished by Symbolics has numerous problems, including various bugs, lack of development environment and debugging facilities, and inadequate support. SDC is committed to a Prolog environment for its natural language work, since its present large system has been written in Prolog. Lack of an adequate Prolog on the Symbolics will present serious problems in developing the system, as well as in integrating smoothly with the Lisp components under development at NYU.

5. Action Required by the Government

Timely negotiation of the Prolog License agreement to ISI would enable SDC to use the Goverment-furnished Symbolics machine.

6. Fiscal Status

(1) Amount currently provided on contract:
$339,728 (funded) $683,105 (contract value)

(2) Expenditures and commitments to date:
$ 86,540 (through August 31, 1985)

(3) Funds required to complete work:
$253,188 (Year 1) $596,565 (Yrs. 1-2)